



**QUEEN'S
UNIVERSITY
BELFAST**

A Review of the Implicit Motion Solver Algorithm in OpenFOAM® to Simulate a Heaving Buoy

Devolder, B., Schmitt, P., Rauwoens, P., Elsaesser, B., & Troch, P. (2015). *A Review of the Implicit Motion Solver Algorithm in OpenFOAM® to Simulate a Heaving Buoy*. Paper presented at 18th Numerical Towing Tank Symposium (NuTTS'15), Cortona, Italy. <http://www.numeca.com/en/about-us/news/numeca-sponsoring-18th-numerical-towing-tank-symposium-nutts-15>

Queen's University Belfast - Research Portal:

[Link to publication record in Queen's University Belfast Research Portal](#)

Publisher rights

© 2015 The Authors

General rights

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

A Review of the Implicit Motion Solver Algorithm in OpenFOAM® to Simulate a Heaving Buoy

Brecht Devolder^{1,*}, Pál Schmitt², Pieter Rauwoens³, Björn Elsaesser², Peter Troch¹

* Corresponding author. Tel.: +32 9 264 54 92; Fax: +32 9 264 58 37; Mail: Brecht.Devolder@UGent.be

¹ Ghent University, Department of Civil Engineering, Technologiepark 904, 9052 Ghent, Belgium (Brecht.Devolder@UGent.be, Peter.Troch@UGent.be)

² Queen's University Belfast, Marine Research Group, BT9 5AG Belfast, Northern Ireland, United Kingdom (p.schmitt@qub.ac.uk, b.elsaesser@qub.ac.uk)

³ KU Leuven, Department of Civil Engineering, Zeedijk 101, 8400 Ostend, Belgium, (Pieter.Rauwoens@kuleuven.be)

1 Introduction

Heaving buoys are currently very interesting with regard to renewable energy. More specific, heaving buoys, in general also called Wave Energy Converters (WECs), can be used to extract wave energy from ocean waves. In order to extract a considerable amount of wave power, large numbers of WECs are arranged in farms.

Prior to the analysis of farm effects, the fluid characteristics around a single WEC have to be understood in detail. Computational Fluid Dynamics (CFD) is able to solve the viscous flow field in three dimensions around a floating object. OpenFOAM® (2014) is selected as a suitable CFD package to investigate the flow field around and the response of a heaving buoy in an incident wave field. OpenFOAM is a robust and advanced open source CFD package. The two phase flow solver with dynamic mesh handling, interDyMFoam, is available in OpenFOAM. Wave generation and absorption are implemented in the IHFOAM toolbox (Higuera, 2013a, 2013b). Being open source, it enables the user to develop a coupling strategy with another far field solver to reduce the computational cost of a simulation of an entire farm.

There are several issues regarding the simulation of floating bodies in a dense fluid with CFD which form the subject of the present contribution. Key issues are related to the convergence of the motion solver and the coupling between the motion and fluid solver.

2 Numerical framework

In the current paragraph, a concise description of the numerical setup is given, focussing on the solvers and toolboxes used for the case studies selected.

2.1 Fluid solver

The two phase interDyMFoam solver, developed for dynamic mesh handling, is based on

the interFoam solver for static meshes. The flow field is calculated using the incompressible Navier-Stokes equations. The solver makes use of the Volume over Fluid (VoF) method to track the interface between the two fluids. The VoF method is an excellent tool in the field of coastal engineering to simulate complex free surfaces deformations, including wave breaking. interDyMFoam combines the VoF method and a mesh deformation solver. The mesh is deformed according to the motion of a rigid body. The motion of the body is determined by a motion solver, introduced in the next section.

The motion of a floating body will generate radiated waves. The wave height dampens out when the wave travels further away from the body. When these radiated waves hit the boundaries of the computational domain, reflection should be avoided. Therefore the IHFOAM toolbox (Higuera, 2013a, 2013b) is used to absorb the waves at the boundaries by a specific boundary condition. The absorption methodology is based on shallow water theory. The validity of the underlying assumption, shallow water, will be evaluated during the numerical simulations. As mentioned in Higuera (2013a), the absorption condition works relatively well for waves outside the shallow water range. However, a careful assessment of the suitability of the method is still desirable.

2.2 Motion solver

For the cases studied here, the buoys are restricted to move solely in the upward and downward direction. Only one degree of freedom is considered, the heave motion, instead of the general six degrees of freedom.

The standard motion solver in OpenFOAM uses a second order accurate leapfrog scheme to calculate the velocity and the position of the object based on the acceleration (Dullweber, 1997). The acceleration is derived from Newton's second law.

2.3 Coupling strategy

The fluid solver and motion solver are coupled to simulate rigid body motions. The coupling is explained by following the methodology inside the `interDyMFoam` solver of OpenFOAM-2.3.1, which is visualised in the flow chart provided in Figure 1. At the start of a certain time step, the motion solver is called first, represented by the dashed box in Figure 1. Different functions are evaluated inside that motion solver. Details about “Update position” and “Update acceleration”, which are based on a leapfrog scheme, are given in paragraph 4.2. In between the “Update position” and “Update acceleration”, the total force (i.e. pressure, viscous, weight and acceleration forces) on the body is calculated. After the new position of the rigid body is determined, the object is moved to its new position (“Move object”). Next, the mesh is deformed and moved (“Move mesh”) where the new position of the rigid body serves as a boundary condition. When the mesh is moved to the new position, the reference mesh is always the one obtained from the previous time step. After the new mesh is obtained, the fluid solver is started. First the field flux is corrected, followed by the VoF method to track the free surface. Thereafter, the PIMPLE algorithm solves respectively the momentum and pressure equations to calculate the velocity field and the pressure field. If the maximum number of PIMPLE iterations is not reached, a new iteration is started within the same time step. Otherwise, a new time step is triggered.

The motion state of the rigid body (i.e. acceleration, velocity, centre of mass) from the old time step is stored when the function “New time” is called to access it during the new time step.

The coupling between the fluid and motion solver is regulated by multiple PIMPLE iterations in every time step. This is because the motion state and the flow field are calculated one after the other. The purpose of these iterations is to obtain a solution where the motion of the object is in equilibrium with the flow field at a certain time step. The implicit iterations between the fluid and motion solver are only implemented since OpenFOAM-2.3.x. The number of iterations is key to faster simulation times. The stronger the coupling, the lower the number of iterations, the faster the simulation speed. Therefore a strong coupling between fluid and motion solver must be achieved.

3 Test cases

Two different case studies are presented to explain some principle ideas regarding the motion and the force on a floating body. The first geometry is a 3D floating buoy which solely operates in heaving mode. Figure 2 visualises a slice of the hexahedral grid structure around the heaving buoy.

Due to the complex mesh structure around a heaving buoy, a simplistic 2D floating body is used as a second test case: a floating block in a two dimensional situation which again only moves in the

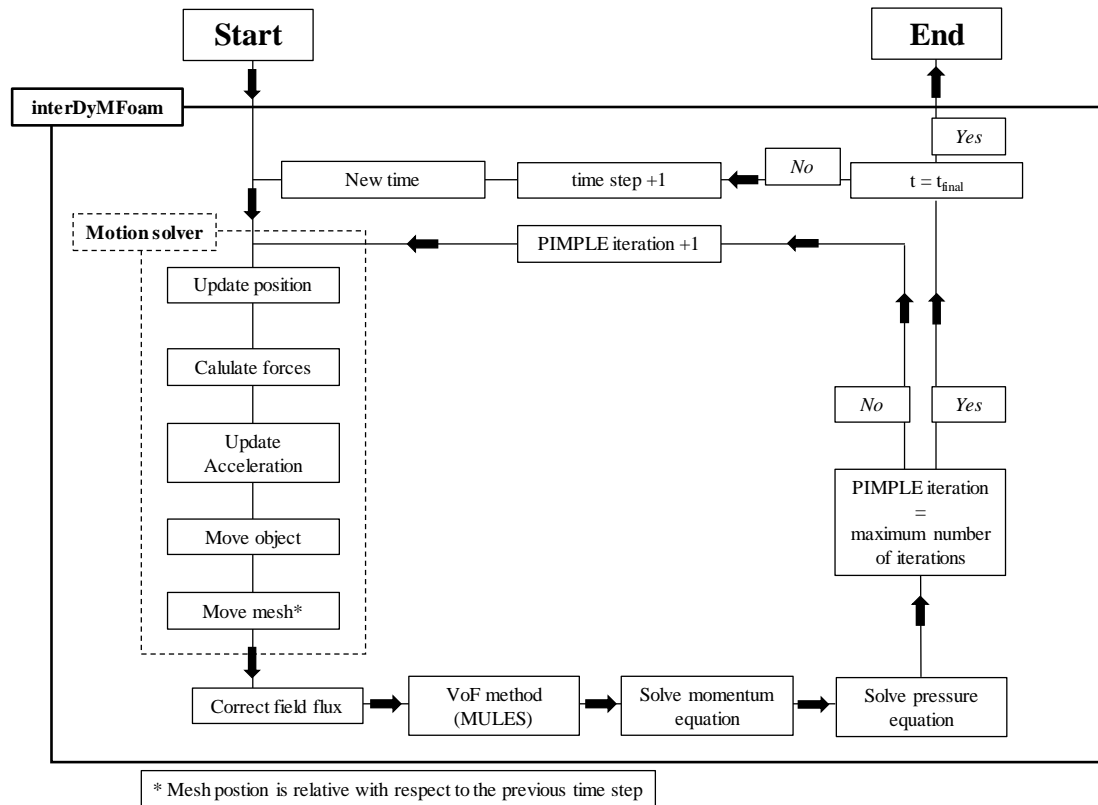


Figure 1: Flow chart of the `interDyMFoam` solver in which a detail of the motion solver is provided.

heave direction. A definition sketch of the second test case is provided in Figure 3. The mesh structure consists of a dense hexahedral Cartesian grid. Compared to the buoy shown in Figure 2, the width of the object in Figure 3 is deliberately chosen larger than the height in order to reveal the deficiencies of the present numerical implementation (see further).

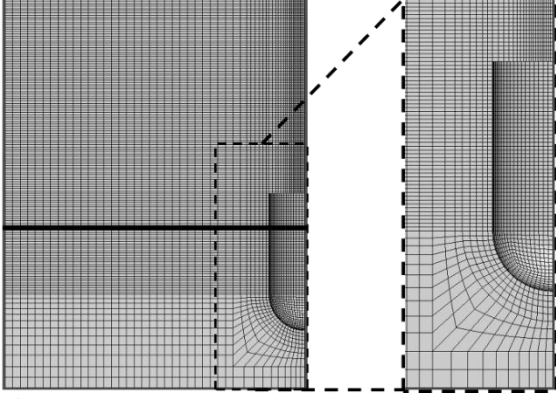


Figure 2: The hexahedral mesh structure around a heaving buoy (3D). The thick horizontal solid black line at the left figure indicates the initial free surface.

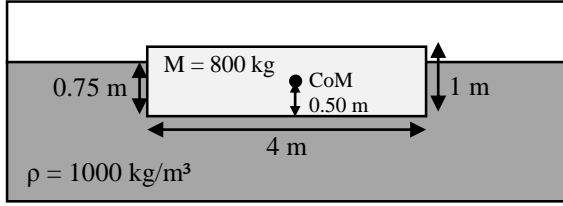


Figure 3: A definition sketch showing the geometry of a 2D floating block ($\rho_{\text{block}} = 200 \text{ kg/m}^3$). The initial draft is 0.75 m and the position of the centre of mass (CoM) is located in the middle of the block.

4 Results and solver optimization

4.1 Force on a floating body

The force on a floating object is determined for a free decay test. In that particular test, the floating object is placed out of equilibrium leading to a damped oscillatory motion until all the forces acting on the object are in equilibrium. Numerical results for the free decay test of a heaving buoy (Figure 2) show spikes in the total force acting on the floating object, as indicated in Figure 4. These spikes are not expected because the total force on the object during a free decay test is theoretically described by a sinusoidal function multiplied by an exponential decay which is per definition a smooth function. The spikes disappear in the graph expressing the position of the centre of mass of the heaving buoy in function of the time, as shown in Figure 5. However, the position is the second derivative of the acceleration which is in itself a linear function of the total force on the floating object.

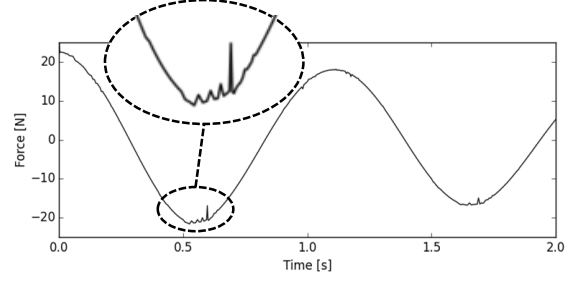


Figure 4: Total vertical force on a heaving buoy (Figure 2) in function of the time. The dashed circle shows a detail of the spikes in the total force.

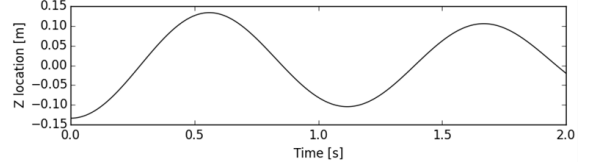


Figure 5: Centre of mass in the Z-direction of a heaving buoy (Figure 2) in function of the time (reference $Z = 0 \text{ m}$ is equal to the initial Still Water Level).

These observations have led to a more profound analysis of the motion solver implemented in OpenFOAM. In the remaining part of the paper, the geometry is changed and simplified from a heaving buoy, Figure 2, to a 2D floating block operating in heave, Figure 3.

4.2 Leapfrog scheme instability

The leapfrog scheme originally programmed in the motion solver consists of three subsequent steps to update the motion state (for the heave motion only):

- Update position:

$$v_{i+1}^{n+1/2} = v_i^n + 0.5 \cdot \Delta T^n \cdot a_i^{n+1} \quad (4-1)$$

$$CoM_{i+1}^{n+1} = CoM_i^n + \Delta T^{n+1} \cdot v_{i+1}^{n+1/2} \quad (4-2)$$

- Calculate the total force acting on the body:

$$f_{Global,i+1}^{n+1} = \sum_j^{body} p_j A_j + \sum_j^{body} \tau_j A_j - m \cdot g \quad (4-3)$$

- Update acceleration:

$$a_{i+1}^{n+1} = \frac{f_{Global,i+1}^{n+1}}{m} \quad (4-4)$$

$$v_{i+1}^{n+1} = v_{i+1}^{n+1/2} + 0.5 \cdot \Delta T^{n+1} \cdot a_{i+1}^{n+1} \quad (4-5)$$

in which $n+1$ is the current time step, $i+1$ is the current PIMPLE iteration, v is the velocity of the body, CoM is the centre of mass of the body, ΔT is the time step, f_{Global} is the total force acting on the body, p_j is the pressure acting on each boundary face

around the body, τ_j is the shear stress acting on each boundary face around the body, A_j is the area of a boundary face, m is the dry mass of the body and g is the gravitational acceleration.

According to Birdsell & Langdon (2004), the leapfrog scheme is stable for a fixed time interval. In order to rule out any problems related to a variable time step, all simulations hereafter are performed using a fixed time step for the entire simulation time.

First, the originally implemented leapfrog scheme is analysed by a free decay test of the case study given in Figure 3, a 2D floating block. The number of PIMPLE iterations is set to 25 and a fixed time step of 0.005 s is used. Due to instability problems, relaxation of acceleration is set to 0.1. A more detailed description regarding acceleration relaxation is given in the section 4.3. Figure 6 and Figure 7 show the total vertical force on the 2D block, respectively as a function of the iterations and as a function of the time. In general, the forces are converging in a certain time step (Figure 6). It means that the body and fluid motion are in equilibrium after 25 PIMPLE iterations. However, the main problem is related to the converged value of the total force in every time step. As shown in Figure 7, large oscillations in the total force on the floating block between the different time steps are observed. Again, these spikes are not expected in the beginning of the simulation because the total force during a free decay test should be smooth and follow a damped cosine function.

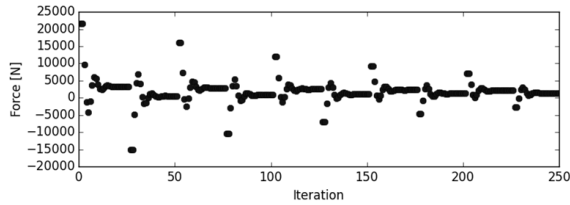


Figure 6: Total vertical force on the 2D block in function of the iterations for a fixed time step of 0.005 s.

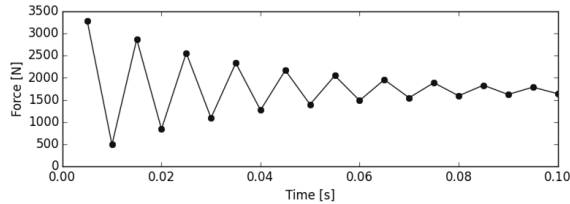


Figure 7: Total vertical force on the 2D block in function of the time for a fixed time step of 0.005 s.

The results presented in this section show that a fundamental problem exists in the implementation of the motion solver. Theoretically, the leapfrog scheme is an explicit scheme. However, equation (4-1), which is originally implemented in OpenFOAM, has an implicit character. This is because the acceleration from the previous iteration of the same time step, a_i^{n+1} , is used to update the

CoM at the current iteration of the same time step, CoM_{i+1}^{n+1} . This is opposite to the theoretical formulation of the leapfrog scheme, which uses the acceleration from the previous time step, a^n (Dullweber, 1997). Therefore equation (4-1) is rewritten to:

$$v_{i+1}^{n+1/2} = v^n + 0.5 \cdot \Delta T^n \cdot a^n \quad (4-6)$$

It means that the leapfrog scheme is made explicit because the acceleration from the previous time step, a^n , is used to calculate the position at the current time step, CoM_{i+1}^{n+1} . This also means that the time consuming fluid solver is only needed once in every time step because the position of the object remains constant in a certain time step.

The implementation of equation (4-6) is checked by using a mock-up fluid solver where in the first instance the force on the object is analytically determined by the upward hydrostatic force and the downward weight of the body:

$$\begin{aligned} f_{Global} &= \rho_w \cdot V_{wet} \cdot g - m \cdot g \\ &= -\rho_w \cdot A_{wet} \cdot g \cdot \Delta z \end{aligned} \quad (4-7)$$

in which ρ_w is the density of water, V_{wet} is the underwater volume of the floating object, A_{wet} is the horizontal water plane area and Δz is the distance between the CoM at time step $n+1$ and the CoM in equilibrium. The first line in equation (4-7) can be rewritten to the second line by some basic geometrical considerations. Newton's second law is used to derive the acceleration of the object:

$$\begin{aligned} a^{n+1} &= \frac{f_{Global}}{m} = \frac{-\rho_w \cdot A_{wet} \cdot g \cdot \Delta z}{m} \\ &= -\Delta z \cdot \frac{\rho_w \cdot g}{\rho_b \cdot h_b} = -\Delta z \cdot k \end{aligned} \quad (4-8)$$

in which ρ_b is the density of the floating block, h_b is the total height of the block and k can be seen as constant value.

Numerical results for the acceleration (eq. (4-8)) of the 2D block are provided in Figure 8. One iteration in every time step is performed. The figure shows the acceleration in function of the time. The progress of the acceleration matches the expectations, starting at a maximum value and going downward without any spikes. It proves that the leapfrog scheme based on equation (4-6) is working correctly without any issues regarding stability or convergence.

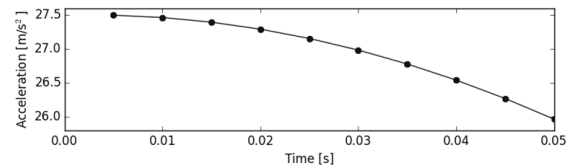


Figure 8: Vertical acceleration of the 2D block in function of the time for a fixed time step of 0.005 s.

4.3 Added mass instability

As mentioned in paragraph 4.1, the simulation of a 2D floating block failed when the width of the object increases with respect to the height. The reason for that phenomenon is probably due to an added mass instability. A possible solution was to use relaxation of acceleration.

In the present implementation, an explicit leapfrog scheme is used inside the motion solver (eq. (4-6)). Starting from the following equation:

$$m \cdot a + m_a \cdot a = -\Delta z \cdot k \cdot m \quad (4-9)$$

in which m_a is the added mass and the right hand side is the force (hydrostatic part and weight of the block) derived from equation (4-8). Compared to equation (4-8), the term $m_a \cdot a$ is added to account for the acceleration force of the fluid on the object. This is a better approximation to the reality than equation (4-8) because all the fluid dynamics are incorporated in equation (4-9), except for the viscous forces (damping forces). Equation (4-9) can be rewritten to:

$$a^{n+1} = -\Delta z \cdot k - \frac{m_a}{m} \cdot a^n \quad (4-10)$$

in which a^{n+1} is the acceleration at time step $n+1$ and a^n is the acceleration from the previous time step. Three different numerical simulations are performed to check the influence of the added mass m_a . Again, only one iteration for every time step is simulated. Figure 9 shows the numerical results for the acceleration (eq. (4-10)) of the 2D block in function of time for respectively $m_a/m = 0.5, 1.0$ and 1.5 . In case $m_a < m$, the oscillation in acceleration damps out. For $m_a = m$, the oscillation remains constant. For $m_a > m$, the oscillation increases and the simulation fails.

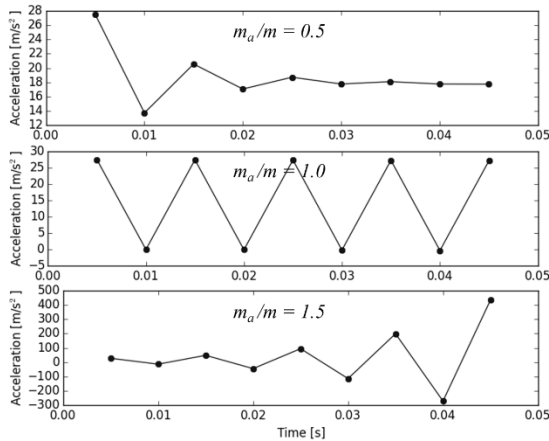


Figure 9: Vertical acceleration of the 2D block in function of the time where $m_a/m = 0.5$ (a), 1.0 (b) and 1.5 (c).

A possible solution to obtain a stable result is to rewrite equation (4-9) to an implicit formulation:

$$a_{i+1}^{n+1} = -\Delta z \cdot k - \frac{m_a}{m} \cdot a_i^{n+1} \quad (4-11)$$

in which a_{i+1}^{n+1} is the acceleration at the current iteration of time step $n+1$ and a_i^{n+1} is the acceleration from the previous iteration in the same time step. Also relaxation of acceleration is needed to reach a converged solution via a stable way in every time step, independent of the value of the relaxation factor. A smart way of applying relaxation exist in literature, related to the added mass effect (Söding, 2001):

$$a_{relax,i+1}^{n+1} = \frac{m \cdot a_{i+1}^{n+1} + m_a \cdot a_i^{n+1}}{m + m_a} \quad (4-12)$$

$$= \alpha \cdot a_{i+1}^{n+1} + (1 - \alpha) \cdot a_i^{n+1}$$

in which m is the dry mass of the body, m_a is the added mass and α is the relaxation factor. The value of the relaxation factor is strongly coupled to the value of the added mass ($\alpha = m_a/(m+m_a)$) and will determine the way how to reach a converged time step. This is explained with Figure 10, Figure 11 and Figure 12 where the acceleration (eq. (4-12)) of the 2D block is given in function of the time or number of iterations. The added mass is set equal to the dry mass of the object ($m_a = m$), a fixed time step of 0.005 s is used and 20 iterations per time step are performed. In Figure 10, the value of the relaxation factor is equal to 0.5, which is exact $m_a/(m+m_a)$. Only one iteration is needed to reach convergence in the acceleration. In case the relaxation factor is 0.75 (Figure 11), convergence of the acceleration is reached with oscillations. This is opposite when a relaxation factor of 0.25 is used (Figure 12). Then, convergence of the acceleration is reached homogeneously without oscillations. The converged value of acceleration in every time step is the same for the three different relaxation factors presented. However, the way to reach convergence over the iterations for a certain time step is different. The same observations are obtained when the added mass increases (e.g. $m_a = 9m$). However, the stability region is narrower which means that the relaxation factor should not deviate too much from $m/(m+m_a)$. It proves that for the method presented, the value of the added mass should be known sufficiently accurate in case of significant added mass effects. Söding (2001) proposes a strategy to calculate the added mass based on a non-linear least squares method to obtain a converged time step after three implicit iterations. The trick is to understand that the total force on the object is dependent on the acceleration, linked by the added mass.

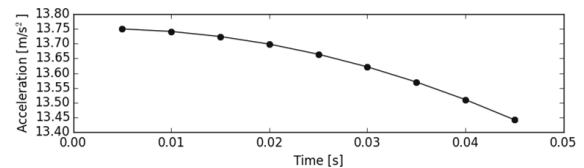


Figure 10: Vertical acceleration of the 2D block in function of the time ($m_a/m = 1.0$, relaxation factor = 0.5).

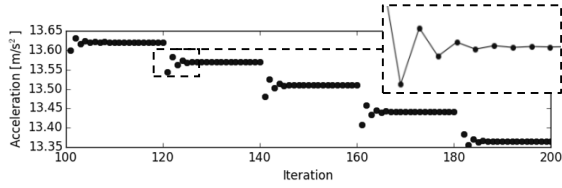


Figure 11: Vertical acceleration of the 2D block in function of the number of iterations ($m_a/m = 1.0$, relaxation factor = 0.75).

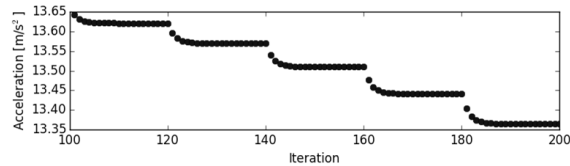


Figure 12: Vertical acceleration of the 2D block in function of the number of iterations ($m_a/m = 1.0$, relaxation factor = 0.25).

5 Research topics under investigation

The coupling between the fluid and motion solver has to be analysed to the finest details. The acceleration must be calculated based on the force obtained with a real fluid solver (eq. (4-3)). The value of added mass should be determined accurately to obtain a stable simulation leading to a converged solution. Söding (2001) could serve as a guideline to calculate the added mass. However, the added mass instability is only significant for a wide object (e.g. Figure 3) but maybe it can be important for a 3D heaving buoy. What will happen if (extreme) waves are added to the numerical model? The motion solver may become unstable for a single heaving buoy. Therefore the presented research aims to develop a general motion solver which operates in all conditions for an arbitrary geometry. With only a relatively small effort, a complete six degrees of freedom motion solver can be developed.

The theoretical leapfrog scheme needs a fixed time step to be stable. However, a time step varying according the Courant number could lead to significant faster simulation times.

For coastal engineering purposes, the radiated wave field at a considerable distance from the buoy is important. However, there are some indications that propagation of radiated waves is a difficult problem in numerical studies using VoF methods. The start point of the propagation is the quality of the generated radiated waves. The quality is directly linked to the performance of the motion solver and the coupling between the motion and fluid solver. When waves are going to be generated at the inlet of the computational domain, two different wave fields are combined. The incident and radiated waves have both a different time and length scale which can be a challenge for a numerical study.

6 Conclusions

The aim of the paper was to present a thorough review of the interDyMFoam solver, especially the motion solver. Some pitfalls in the implemented methodology came up and were described. A new implementation has been presented and used to describe an academic case study of a 2D floating block.

The paper presented is a trigger to develop a stable motion solver for an arbitrary object. The added mass effect should be included. An introduction to the added mass instability was presented. A fast converging methodology is found in Söding (2001) which seems to be worth to investigate within OpenFOAM. A successful implementation would lead to a low number of implicit iterations, minimal three, together with larger time steps.

Acknowledgements

The Research Foundation – Flanders, Belgium (FWO) is gratefully acknowledged for the funding grant.

References

- Birdsall, C. K., & Langdon, A. B. (2004). *Plasma Physics via Computer Simulation*. Taylor & Francis.
- Dullweber, A., Leimkuhler, B., & McLachlan, R. (1997). Symplectic splitting methods for rigid body molecular dynamics. *The Journal of Chemical Physics*, 107(15), 5840–5851.
- Higuera, P., Lara, J. L., & Losada, I. J. (2013a). Realistic wave generation and active wave absorption for Navier-Stokes models. Application to OpenFOAM. *Coastal Engineering*, 71, 102–118.
- Higuera, P., Lara, J. L., & Losada, I. J. (2013b). Simulating coastal engineering processes with OpenFOAM. *Coastal Engineering*, 71, 119–134.
- OpenFOAM®. (2014). <http://www.openfoam.org/>
- Söding, H. (2001). How to Integrate Free Motions of Solids in Fluids. In *4th Numerical Towing Tank Symposium*. Hamburg, Germany.